Learn Java: Hello World

Print Line in Java

In Java, `System.out.println()` can print to the console:

- `System` is a class from the core library provided by Java
- `out` is an object that controls output
- `println()` is a method associated with that object that receives a single argument

```java
System.out.println("Hello, world!");
// Output: Hello, world!
```

`main()` Method in Java

In Java, every application must contain a `main()` method, which is the entry point for the application. All other methods are invoked from the `main()` method.

The signature of the method is `public static void main(String[] args) { }`. It accepts a single argument: an array of elements of type `String`.

```java
public class Person {

  public static void main(String[] args) {

    System.out.println("Hello, world!");

  }

}
```

Java Classes

In Java, a class represents a single concept.

A Java program must have one class whose name is the same as the program filename.

In the example, the `Person` class must be declared in a program file named **Person.java**.

```java
public class Person {

  public static void main(String[] args) {

    System.out.println("I am a person, not a
computer.");

  }

}
```

Statements

In Java, a statement is a line of code that executes a task and is terminated with a `;`.

```java
System.out.println("Hello World");

System.out.println("Java Programming Language");
```

Comments in Java

In Java, comments are bits of text that are ignored by the compiler. They are used to increase the readability of a program.

Single line comments are made by using `//` and multi-line comments are made by starting with `/*` and ending with `*/`.

```java
// I am a single line comment!

/*
I am a
multi-line
comment!
*/
```

Whitespace in Java

Whitespace, including spaces and newlines, between statements is ignored.

```java
System.out.println("Example of a statement");

System.out.println("Another statement");

// Output:
// Example of a statement
// Another statement
```

Compiling in Java

In Java, when we compile a program, each individual class is converted into a **.class** file, which is known as byte code.

The JVM (Java virtual machine) is used to run the byte code.

```
# Compile the class file:
javac first.java

# Execute the compiled file:
java first
```